

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1	Before You Begin	1-1
1.1.1	Responsibility Distribution	1-1
1.1.2	What is Available and Not from SEAPR for a Translation	1-2
1.2	Equipment Requirements.....	1-2
1.3	How to Use This Book.....	1-3
1.4	Conventions	1-3
1.5	Related Documentation.....	1-4
1.6	Release History	1-5
1.7	Terminology	1-5
1.8	Product Overview	1-5

Chapter 2 - Installation

2.1	Installation Procedures.....	2-1
2.2	Shut-down Procedures	2-1
2.3	Start-up Procedures.....	2-1

Chapter 3 - Configuration/Application Building

3.1	Introduction.....	3-1
3.1.1	NLS System in AdvaCommand for Windows NT	3-2
3.2	Adding a New Language	3-3
3.2.1	Setting the Language	3-3
3.2.2	New Directories	3-4
3.2.3	Translate All Standard Texts.....	3-5
3.2.4	Message Catalogs and Header Files for the Translated Texts	3-5
3.2.5	System Variable files	3-6
3.2.6	Keyboard Mapping	3-7
3.2.7	Font Handling	3-9
3.2.8	Resource Files.....	3-11
3.2.9	Alarm and Event Handling	3-11
3.2.9.1	Process Alarm and Event Texts.....	3-11
3.2.9.2	System Alarm Texts	3-15
3.2.10	Printer Handling.....	3-19
3.2.10.1	Line Printer	3-19
3.2.10.2	PostScript	3-19
3.2.10.3	Hardcopy	3-20
3.2.11	RTA- Board Backup.....	3-20
3.2.12	Station Backup.....	3-20

CONTENTS (continued)

- 3.3 How to Add and Modify Texts..... 3-20
 - 3.3.1 Display Builder 3-22
 - 3.3.1.1 Display Description 3-22
 - 3.3.1.2 Dynamic Keys 3-25
 - 3.3.1.3 Display Element 3-27
 - 3.3.2 Process Object Description 3-30
 - 3.3.3 Display Element and Process Dialogs..... 3-31
 - 3.3.3.1 Standard Display Element NLSPD01 3-31
 - 3.3.3.2 Changes in Display Elements..... 3-31
 - 3.3.3.3 Changes in Process Dialogs 3-34
- 3.4 Configuration/Application Building Menus 3-35

Chapter 4 - Runtime Operation

- 4.1 Operating Overview 4-1
- 4.2 Runtime Operation Menus 4-1

Chapter 5 - Maintenance

- 5.1 Backup/Restore Procedures 5-1

Appendix A - Text Files

- A.1 dcBasic A-1
- A.2 dcDialogMsg A-4
- A.3 dcMMITEXT A-5

INDEX

Chapter 1 Introduction

1.1 Before You Begin

The purpose of this document is to give an overall description of the AdvaCommand for Windows NT, localization functionality. The intended audience is the software developers who are going to add a new language to the AdvaCommand for Windows NT. You have to have some knowledge of the AdvaCommand for Windows NT and the terminology used within the business area. You must decide who will be responsible for the work to be done and who will take the maintenance responsibility for the translated version.

1.1.1 Responsibility Distribution

The basic software is developed and owned by ABB Automation Products AB (SEAPR). The responsibility for maintenance and development of English, German and Swedish languages lies within SEAPR. The responsibility for developing of translated texts in the software lies at the translating LIAD. The responsibility for maintenance and further development of texts lies at the LIAD.

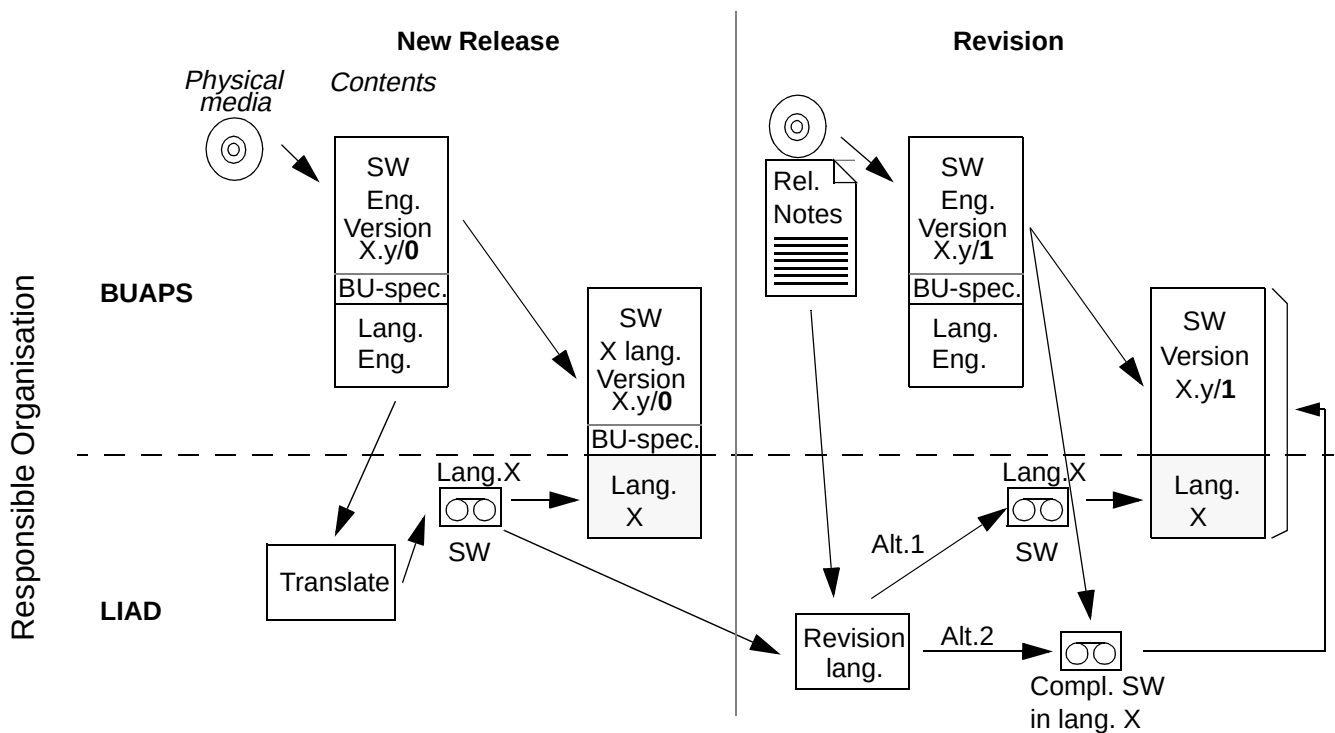


Figure 1-1. Responsibility borderline for translation of texts in the AdvaCommand for Windows NT

1.1.2 What is Available and Not from SEAPR for a Translation

From SEAPR you can order a CD with the standard product software. The standard product supports the languages English, German, and Swedish. For how to make a translation there is a manual (this). The manual describes the procedure how to adapt the software to the local language.

Some localized hardware can not be ordered from SEAPR. This hardware has to be ordered from the local PC purchaser. PC keyboard, if not possible to use any of the above mentioned languages, has to be purchased locally. If the printer require some local fonts, the printer has to be purchased locally. A lot of fonts are supported, but to get to know what font sets that are supported you have to check in the product software. Business Unit (BU)-specific additions are not in the standard operator station software. These are normally added by a BU. If there are any questions about the BU-specific part, the BU has to be contacted.

1.2 Equipment Requirements

To make a translation of texts in AdvaCommand for Windows NT you need the software version 1.7. This version requires minimum 64 Mbyte of primary memory to run properly. If the language requires special fonts that are not supported in the AdvaCommand for Windows NT, these fonts have to be requested from the PC vendor. To get localized operator keyboard, you have to ask the PC vendor.

The eXceed/W software includes keyboard files that generally correspond to keyboard layouts used in a country or geographic region. The following 19 keyboard files are supplied:

- Belgian
- Danish
- Dutch
- French Canadian
- French
- German
- Italian
- Latin American
- Norwegian
- Portuguese
- Spanish
- Swedish
- Swiss French
- Swiss German
- UK English 101 keys
- UK English 102 keys
- US English
- DEC LK450/LK250 ¹
- DEC LK450/LK250 Emulation ²

1. Maps DEC's LK450 or LK250 US keyboard when running in DEC LK201 mode.

2. Emulates DEC's LK450 or LK250 US keyboard when running in DEC LK201 mode on a standard 101-key keyboard.

1.3 How to Use This Book

This book is organized into chapters and sections as shown in Figure 1-2.

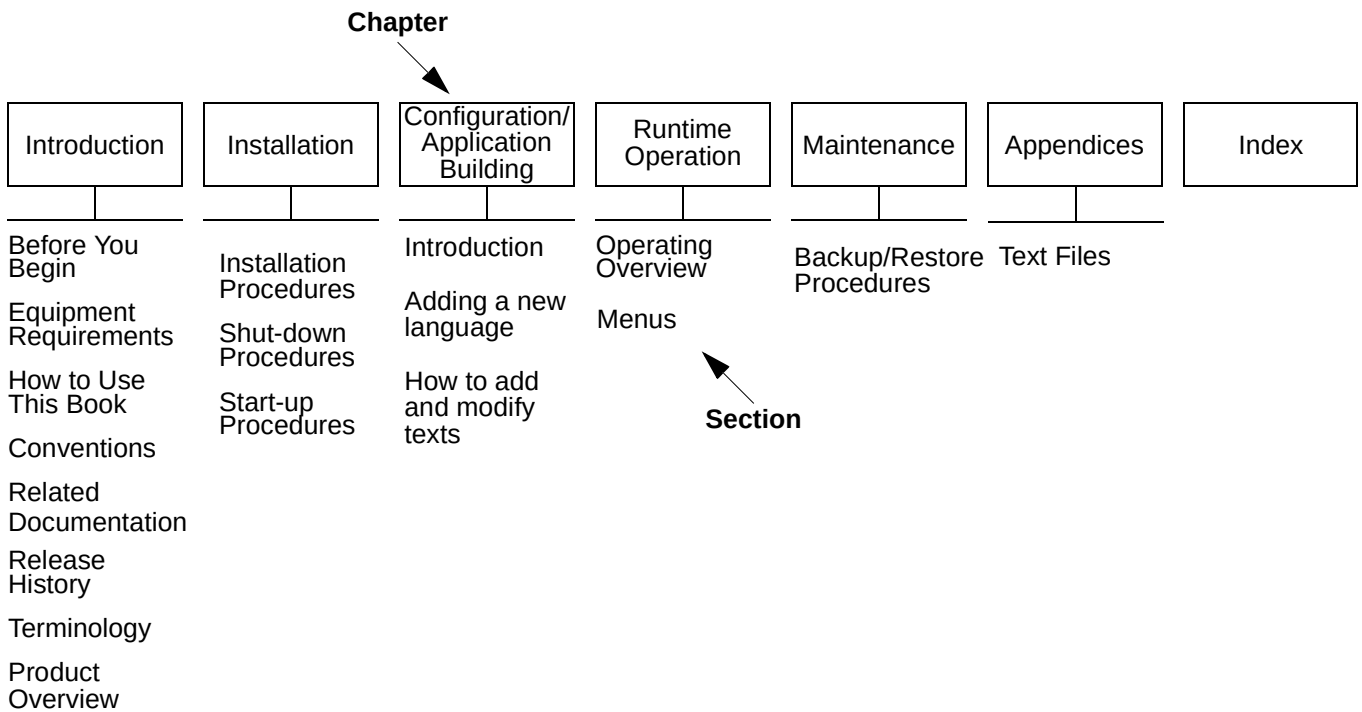


Figure 1-2. Organization of this Manual

1.4 Conventions

The following conventions are used throughout this manual for the presentation of material:

- The words in proper names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- The names of keyboard keys are boldfaced.
 - Initial capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, the **Return** key or <**Return**>.
 - Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
 - In descriptions, the word “key” is used in conjunction with its name. For example, pressing the **Return** key initiates the process.
 - In tables and procedures, angle brackets replace the word “key.” For example, press <**Return**>.
 - A dash is used to indicate that you must simultaneously press several keys. For example, <**Ctrl-C**> indicates that you must hold down the **Control** key while pressing the **C** key.

- The phrase “press and release” is used to indicate that you sequentially press several keys. For example, to close a window, press and release <Alt>, and then <F4>. This indicates that you press and release the **Alt** key, and then you press and release the **F4** function key.
- The names of push and toggle buttons are boldfaced.
 - Initial capital letters are used for the name of a push or toggle button labeled on the display. For example, the **OK** button.
 - In descriptions, tables and procedures, the word “button” is used in conjunction with its name. For example, pressing the **OK** button initiates the process.
- The names of menus and menu items are boldfaced.
 - Initial capital letters are used for the name of menus listed in the menu bar. For example, the **File** menu.
 - Initial capital letters are used for the name of menu items. For example, to open a new window, choose **New** from the **File** menu.
- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

Entered value is not valid. The value must be 0-30.

In a tutorial, you may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

TIC132

1.5 Related Documentation

[Table 1-1](#) lists all documentation related to the AdvaCommand for Windows NT, version 1.7.

Table 1-1. Related Documentation

Category	Title	Description
Hardware	AdvaCommand for Windows NT Peripheral Hardware Reference Manual	This book provides instructions for peripheral hardware setup and installation for the AdvaCommand for Windows NT
Software	AdvaCommand for Windows NT Basic Functions User's Guide	This book give a detailed description of all basic functions in AdvaCommand for Windows NT, including User Interface, Event and Alarm, and System Status.
	AdvaCommand User Interface Reference Manual	This book give a detailed description of all basic display elements and dialogs in AdvaCommand.

Table 1-1. Related Documentation (Continued)

Category	Title	Description
Options	AdvaBuild On-line Builder Reference Manual	This book provides instructions for accessing, and using all On-line Builder functions.
	AdvaCommand for Windows NT Process Sectioning User's Guide	This book provides instructions for accessing, and using all Process Sectioning functions.
	AdvaCommand for Windows NT Status List User's Guide	This book provides instructions for accessing, and using all Status List functions.
	AdvaBuild Display Builder	This book provides instructions for accessing, and using all Display Builder functions.

1.6 Release History

This is the first release of the AdvaCommand for Windows NT software for localization.

1.7 Terminology

The following is a list of terms, that you should be familiar with, associated to the Advant OCS System with ABB Master software. Some terms and expressions used in this document are explained beneath:

Term	Description
NLS	Native Language Support
Internationalization	The design and modification of software so it can be adapted to various local customs and coded character sets.
Localization	The process of adapting an existing software product for use in a particular country or geographic region.

1.8 Product Overview

The product AdvaCommand for Windows NT, version 1.7, is available for translation into other than above mentioned languages.

Chapter 2 Installation

2.1 Installation Procedures

The PC Workstation is delivered with the AdvaCommand version 1.7 for Windows NT software already installed.

For installation of AdvaCommand version 1.7 for Windows NT on a PC Workstation, refer to the *Release Notes* that is delivered with the product versions.

For a description of how to connect the DAT-drive to the PC Workstation, see the appropriate hardware manual (*AdvaCommand for Windows NT Peripheral Hardware Reference Manual*) for information.

2.2 Shut-down Procedures

Refer to *AdvaCommand for Windows NT Basic Functions User's Guide* for information.

2.3 Start-up Procedures

Refer to *AdvaCommand for Windows NT Basic Functions User's Guide* for information.

Chapter 3 Configuration/Application Building

3.1 Introduction

Users of ABB Advant Open Control System (OCS) speak a variety of languages and observe many different cultural practices. Local-language, processing capability is becoming high priority with the kinds of software products in use throughout the world. For this reason, users need software which will easily accommodate local conventions.

The two processes in the NLS approach that enhance software for international use are:

- **Internationalization**

Internationalizing software include:

- Supporting the letters and symbols required to read and write the user's language, for example using standardized character sets that includes the most commonly used languages.
- Processing characters and text according to the rules of the user's language, for example the collation order. Countries have different rules for sorting words. In Spanish, the sequence "ch" and "ll" are treated as single character and are sorted between "c" and "d", and between "l" and "m" respectively.
- Providing the schema to process translated messages and prompts, that is separate the application from the language-dependent information as text in menus, labels, prompts, help, and so on.
- Changing functions and conventions to comply with local requirements. For example representation of data and time format, number formatting and so on.

- **Localization**

Localizing adapts the software to a particular locale, including the translation of messages.

3.1.1 NLS System in AdvaCommand for Windows NT

The NLS system in AdvaCommand for Windows NT has a hierarchical structure in three levels. The levels are subsystem, components and messages.

Each message has a unique identification, consisting of three integers representing Message number, Component number and Subsystem number.

For each subsystem there is a component definition file. The file bsSubSystem.def specifies where to find this file.

For each component there is one file containing all the messages for the component.

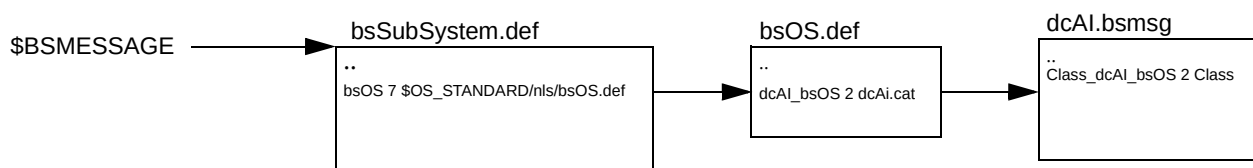


Figure 3-1. Hierarchical structure of the NLS system in AdvaCommand for Windows NT

The identification for messages is: <Message>_<Component>_<Subsystem>

3.2 Adding a New Language

This section describe the preparations that has to be done to add a new language. Follow the sections in sequential order.

3.2.1 Setting the Language

1. First you must set the language in your machine.

This is done from Regional Settings in the Control Panel.

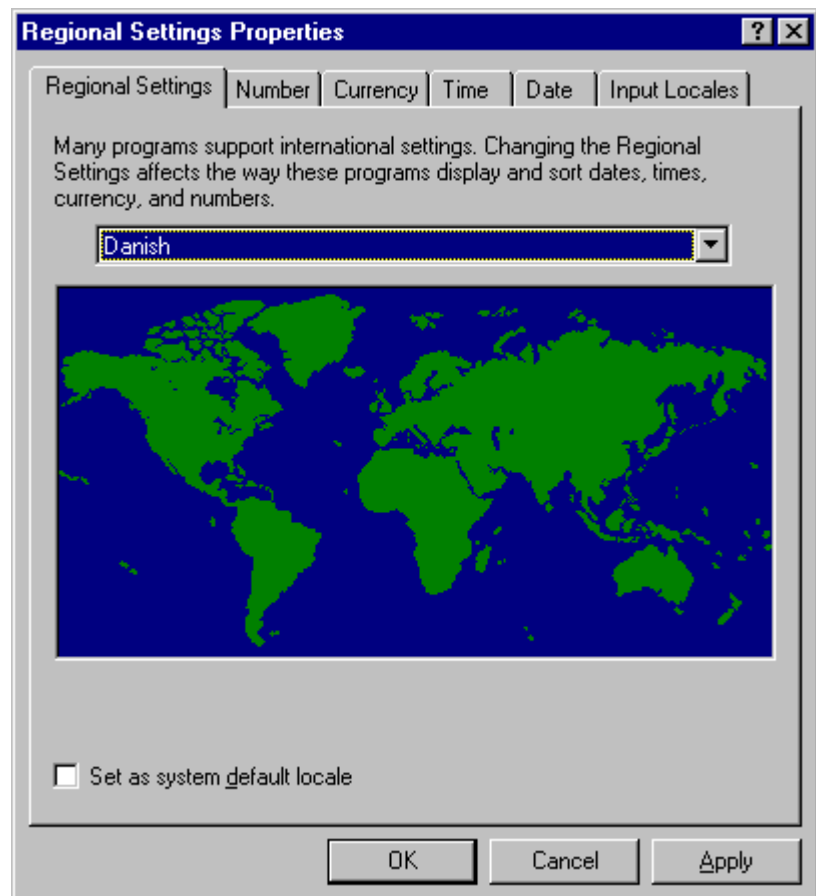


Figure 3-2. Regional Settings

2. Choose your language for example Danish and press OK.

3. You also have to find out the three letter abbreviation for your specific language.

One way to do that is to run the command *bsMessageInstall.exe -v* from the MS-DOS Comman Prompt. You will find the abbreviation on the second row, for example for Danish the row will look like *Using Language DAN* and it is DAN we wanted to know. When we refer to \$LANG you should use the abbreviation you got from *bsMessageInstall.exe -v*.

3.2.2 New Directories

Configuration data is stored in different directory structures depending on who the configurator is, that is who created the configuration data. This also include the NLS files. At delivery NLS files are stored in the directory structure, 'standard'. The directory structures for 'adaption' and 'customer' includes the same sub-directories as 'standard'. Other ABB companies that adds texts to an operator station with AdvaCommand for Windows NT software uses the directory structure 'adaption' for storage of its NLS files. Texts changed or added that affects only the current operator station are stored in the directory structure for 'customer'.

Directory Structure for message files in an operator station with AdvaCommand for Windows NT software:

```
/products/data/standard/OS/nls/$LANG/  
/products/data/adaptions/OS/nls/$LANG/  
/products/data/customer/OS/nls/$LANG/.
```

NOTE

\$LANG is the value of the environment variable LANG. For American-English \$LANG = en_US.iso88591. This format is only used for American-English, German and Swedish.

The directory structure /products/data/standard/OS/nls contains standard NLS files for the Operator Station. This directory structure is only used by ABB Automation Products AB (SEAPR).

Directory structure /products/data/adaptions/OS/nls may be used by other ABB companies for their additions.

Directory structure /products/data/customer/OS/nls is used by the customer.

The directories OS/nls/ contains one directory for each loaded language. The available languages to load in AdvaCommand version 1.7 for Windows NT:

Table 3-1. Available languages

Language	Symbolic name
American-English	en_US.iso88591
German	de_DE.iso88591
Swedish	sv_SE.iso88591

Create the new directories:

1. Create a new directory \$LANG under
 /products/data/adaptions/OS/nls/
 /products/data/customer/OS/nls/
2. Create a new directory \$LANG under
 /products/data/adaptions/OS/lib/
 /products/data/customer/OS/lib/
3. Create a new directory app-defaults under
 /products/data/adaptions/OS/lib/\$LANG/
 /products/data/customer/OS/lib/\$LANG/

Example

If you want to add danish to your operator station the directory structure should look like this:

```
/products/data/adaptions/OS/nls/DAN
/products/data/customer/OS/nls/DAN
/products/data/adaptions/OS/lib/DAN
/products/data/customer/OS/lib/DAN
/products/data/adaptions/OS/lib/DAN/app-defaults
/products/data/customer/OS/lib/DAN/app-defaults
```

3.2.3 Translate All Standard Texts

All texts for the local language resides in message files having the file extension .err. To localize the operator station they have to be translated to the local language. Some of the texts are described in [Appendix A, Text Files](#).

1. Copy the message files from /products/data/standard/OS/nls/en_US.iso88591/ to
 /products/data/adaptions/OS/nls/\$LANG/
2. Translate all texts in the message files under /products/data/adaptions/OS/nls/\$LANG/

NOTE

During the translation to a new language it is important that the new texts are not longer than the English texts.

3.2.4 Message Catalogs and Header Files for the Translated Texts

For each message file a message catalog and a header file is pointed out. Message catalogs are stored under:

```
/ABB/Advant/Data/BasicServices/Rts/$LANG/0
```

and the header files are stored under:

```
/products/data/standard/OS/include/
/products/data/adaptions/OS/include/
/products/data/customer/OS/include/.
```

Message catalogs are used to separate the text from the application code. A message file is used to create this message catalog. To create the message catalog and the header file the message file is compiled with the command:

```
dcNLS messagefile
```

Before running dcNLS you must export NT_LANG to your chosen language, for example export NT_LANG=DAN

Create the new message catalogs and header files:

1. Open a Terminal Window and do the command sh (this will get you into korn shell)
2. Export the NT_LANG variable to your chosen language.
3. Go to the directory /products/data/adaptions/OS/nls/\$LANG/
4. Give the command dcNLS for each message file.

3.2.5 System Variable files

The language affect several parts in the system. See [Table 3-2](#).

Table 3-2. Language

LANGUAGE	LANG	C_LANGUAGE
American-English	en_US.iso88591	01
Swedish	sv_SE.iso88591	46
German	de_DE.iso88591	49

LANGUAGE: Inform which language the data is valid for. LANGUAGE is visible in the Station Settings option menu for System Language.

LANG: This is the value of the environment variable LANG. The value of LANG for the different countries is defined in POSIX.

C_LANGUAGE: The value of the environment variable C_LANGUAGE. C_LANGUAGE is given as country number (tel). C_LANGUAGE is used at the RTA-level. The language on RTA-level can only be set to English, German or Swedish.

System variables is used to realize the table. There is one system variable file for each language with the name NLSVariables.\$LANG. The system variable file for American-English has the name NLSVariables.en_US.iso88591 and contains the following lines:

```
Uis.SystemVariable.dcLanguageText:      "American-English"  
Uis.SystemVariable.dcSysLanguage:       "en_US.iso88591"  
Uis.SystemVariable.dcCLanguage:         "01"  
Uis.SystemVariable.dcKbdMap:            "abb_modkeys.en_US.iso88591"
```

In the option menu for System and Keyboard language the dcLanguageText is presented. dcSysLanguage defines \$LANG and dcCLanguage define \$C_LANGUAGE.

The system variable files are placed under OS/config/.

Create the new system variable file:

1. The name on the new system variable file must be NLSVariables.\$LANG
2. Put the file under /products/data/adaptions/OS/config/.

3.2.6 Keyboard Mapping

When translating from one language to another the keyboard mapping changes as well. For example on a Swedish keyboard is | mapped as ö, \ mapped as Ö, { as ä, [as Å, } as å, and finally] as Ä.

To do this remapping of the keyboard you have to enter Keyboard Properties in the Control Panel. In this window you can Add your language and set a specific language to default.

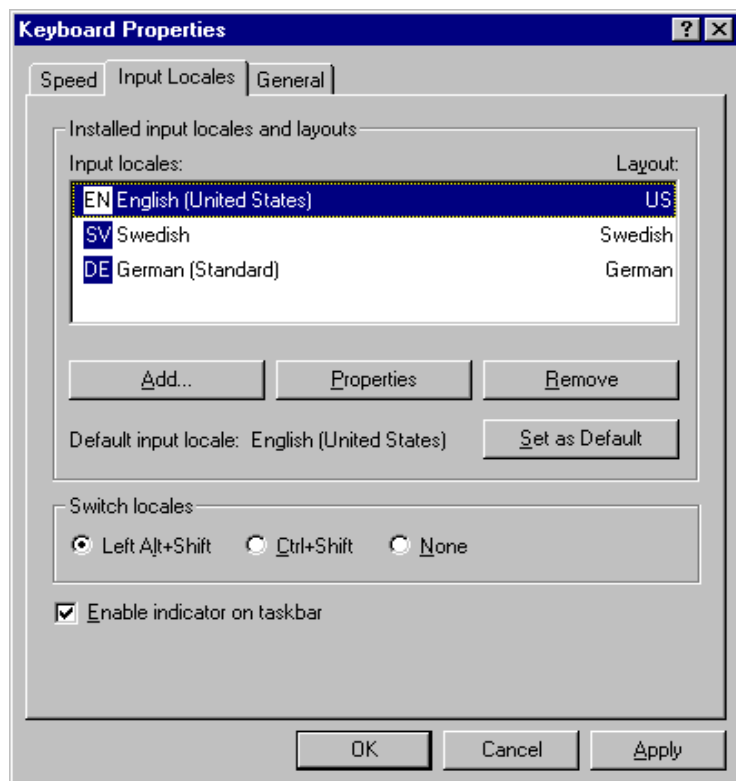


Figure 3-3. Keyboard Properties

You also have to change the keyboard mapping in eXceed/W. Open Xconfig under the eXceed/W folder and double-click on Input.

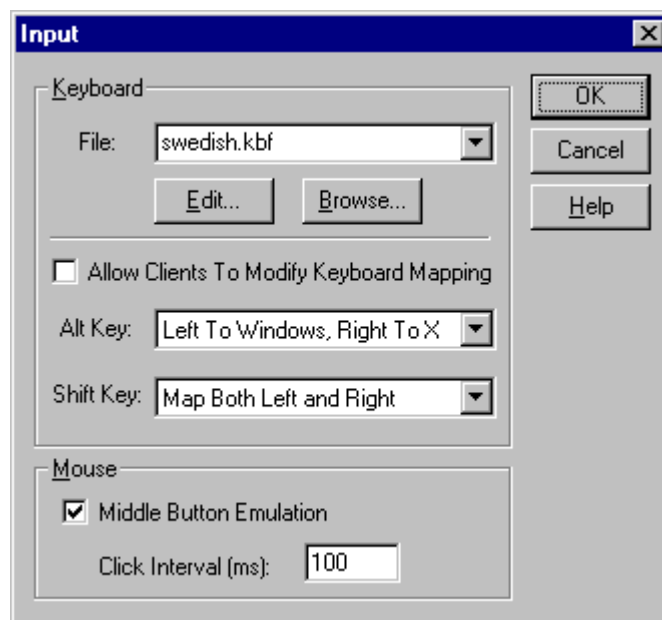


Figure 3-4. eXceed/W keyboard input

Then choose your keyboard mapping from the pulldown menu and press **OK**.

3.2.7 Font Handling

All fonts used in the AdvaCommand version 1.7 for Windows NT are supplied by the eXceed/W X server software. The used fonts can be found in the Font Database in eXceed/W. The Font Database is located under Font in Xconfig and you find Xconfig under Start - Program - Exceed.

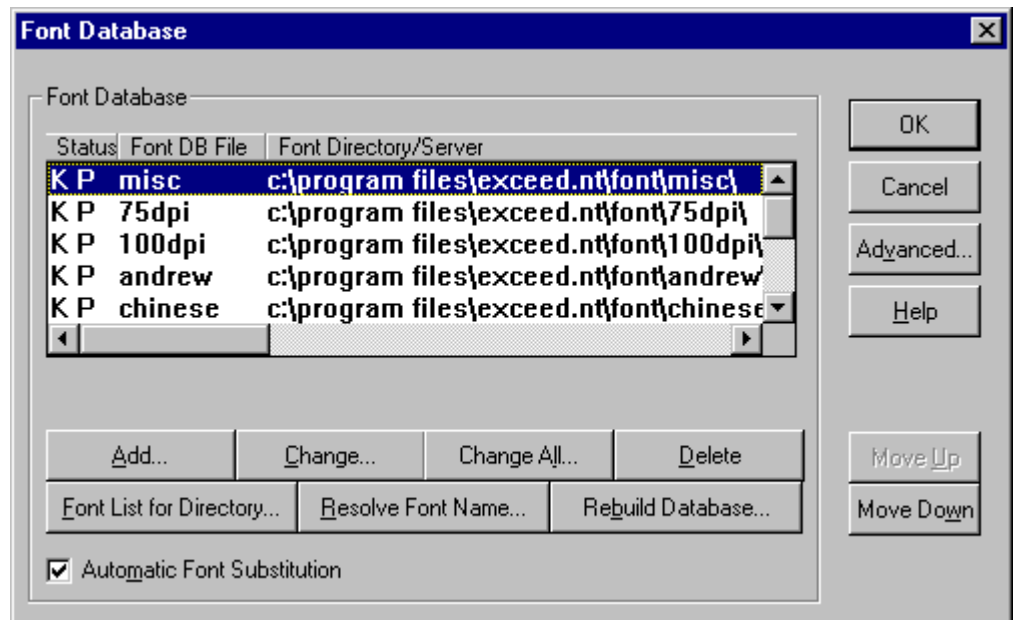


Figure 3-5. Font Database

To get a formatted list of all misc fonts in the X-server you select the Font DB File misc and then you press **Font List for Directory**.

This command will give a list describing all available fonts in the X-server that look like this.

foundry weight set width points vertical resolution average width
↓ ↓ ↓ ↓ ↓
-adobe-new century schoolbook-medium-i-normal--11-80-100-100-p-60-iso8859-1
↑ ↑ ↑ ↑ ↑
font family slant pixels horizontal resolution spacing character set

Figure 3-6. Available fonts in the X-server

For more information about the individual fields, see [Appendix A, Text Files: Specifying Fonts in the XLib Programming Manual](#) (ISBN 1-56592-002-3).

In the operator station with AdvaCommand version 1.7 for Windows NT, fonts are specified with a short font specification consisting of two letters FA-FN and two digits 10-60 in step of 10. These font specifications are connected to real X fonts through a definition file.

The font definition file is placed in the directory:

/products/data/standard/OS/config/

and have the name fontdef.dat.

```
default raster 11 -adobe-courier-bold-r-normal--*-*-*-*-*-iso8859-1
/* FM 10:12 20:14 30:17 40:18 50:24 60:34 Courier-bold-r */
```

The first line specifies that font number 11 shall be a courier font. The font specification is the standard X font specification described above. The second line is used by process graphics and the display builder. FM says that this is font FM, 10:12 specifies that pixelsize 12 shall be used for FM10. The last font name string is only shown in the display builder only.

It is also possible to add a font set used for more than one language by adding a comma separated list of fonts. The specification should be on one line, or as in the example below that uses a line continuation character as the last character on the line.

```
default raster 11 -adobe-courier-bold-r-normal--*-*-*-*-*-iso8859-1,\
                  -adobe-courier-bold-r-normal--*-*-*-*-*-iso8859-5
/* FM 10:12 20:14 30:17 40:18 50:24 60:34 Courier-bold-r */
```

The example above specifies two fonts for the default raster 11 font in Sherrill-Lubinski, one used for languages using iso8859-1 encoding and one used for languages using iso8859-5 encoding.

NOTE

When more than one font is specified the iso8859-1 encoding must be specified first as in the example above.

If the encoding is not iso88591 create a new font definition file:

1. Copy fontdef.dat from /products/data/standard/OS/config/ to /products/data/adaptions/OS/config/
2. Add the new font set after the iso88591 font set in fontdef.dat. Separate the font sets with a comma.

3.2.8 Resource Files

Fonts are also specified in resource files for Motif. The main resource file is file OS in the directory /products/data/standard/OS/lib/en_US.88591/app-defaults. This directory also contains resource specifications for the tools running in the operator station.

Fonts are specified as X-resources, like Application.font: FA10. Do not specify a font too general, like *Font. This may make applications within the operator station hard or impossible to use.

When translating to a new language a new resource file can be added to the directory: /products/data/standard/OS/lib/'new_language'/app-defaults/

The resource files of interest are OS, Dbuild, DisplayTool and NLSEditor.

Dbuild is the resource file for the Display Builder, **DisplayTool** is the resource file for the Display Installer, **NLSEditor** is the resource file for the System List Translator and **OS** is the common resource file for the Operator Station. In these files all fonts have the short font specification from the font definition files. An example from **OS**:

OS*fontList:	FA20
OS*XmPushButton*fontList:	FC20
OS*XmPushButtonGadget*fontList:	FC20
OS*XmLabel*fontList:	FA40

NOTE

In most cases it is not necessary to do anything in the resource files. Since it is possible to define several encoding in the font definition file, one for each language.

If you want to change font for a certain application:

1. Copy the resource file for that application from
/products/data/standard/OS/lib/en_US.iso88591/app-defaults to
/products/data/adaptions/OS/lib/\$LANG/app-defaults
2. Edit the resource file.

3.2.9 Alarm and Event Handling

3.2.9.1 Process Alarm and Event Texts

A process alarm or process event is presented by its name, description, property text and event text in a list. The translation to local language of the property and event text are described in this section.

The property and event texts can be either **user defined** or **standard**. The user defined property and event texts are stored in database elements of type EVENT_TREAT. The standard property and event texts are stored in the database concept TEXT. Both are located on the Real Time Accelerator Board (RTA).

User defined property and event text

The user defined property and event texts for standard event handling are stored in EVENT8 - EVENT14, EVENT16 - EVENT18.

The property and event text for the user defined event handling are stored in EVENT20 - EVENT300.

1. Enter the **System Configuration Dialog**. Select <Maintenance...> and thereafter <**RTA Board Config**>
2. Load the command set for database handling:
 - locpset CMDS:DEAR43A0.CT <CR>
3. Dump the source code for the EVENT_TREAT elements to the volume SRCE: which corresponds to the directory /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE in the target system.
 - DUDBS SRCE:EVENT,SEL=EVENT8 - EVENTX <CR>where X is the number of the last EVENT element with defined property or event text that shall be translated to local language.

The number of created files depends on how many EVENT_TREAT elements that have been dumped. The created files are named EVENTxx.BA.

4. Edit the source code files by using a standard text editor:
 - Enter the **System Configuration Dialog**. Select <AdvaBuild...> and thereafter <**Text Editor**>. The files are found in directory /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE.
5. Load the modified source code files:
 - TRDBS SRCE:EVENTxx.BA <CR>
6. Take a backup of the RTA Board by selecting <**RTA Board Backup**> in the System Configuration Dialog.
7. Make a Function Restart.

Standard property and event text

Translation to a native language shall be made in a logical dump of the TEXT concept. You must make a logical dump manually as follows:

1. Enter the **System Configuration Dialog**. Select <Maintenance...> and thereafter <**RTA Board Config**>
2. Enter in RTA window
 - DUDB SRCE:<file name> for example DUDB SRCE:STANDARDMax length of the file name is 8 characters. Answer the questions as follows:
 - DUMP FROM LF = 31
 - INCLUDE DYNAMIC PROPERTIES? N

- INCLUDE STRUCTURE OF DYNAMIC PROPERTIES? N
- INCLUDE ACTION PROPERTIES? N
- MESSAGE: Dump of standard property and event text
- DUMP START AT RECORD = 2
- DUMP STOP AT RECORD = 278
- ENTER PROPERTY SELECTION DIALOG? N

The dump is stored in /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE/. Below is an extract of the dump. The dump starts with some headlines which must not be changed. The text to be translated starts with line

][2 0 0 0 5 Value 0

where 2 is the record number, 0 0 0 are pointers which shall not be changed. 5 is the number of characters in the property or event text. Thereafter follows the text to be translated. The text and the number of characters can be changed using a standard text editor.

3.][A 201
SRCE:STANDARD.LD
Midas Data Base Logical Dump
Copyright ASEA

58 Selective dump of logfile 31 created 1998-09-10 13:06:20

14 standard texts

5 *+/-/

0 31 0 FFFFFFFF 5

0 2 0 20 0 FFFFFFFF 1

0 3 0 20 0 FFFFFFFF 1

0 5 0 20 0 TFFFFFFF 1

0 6 0 40 0 FFFFFFFF 1

0 7 0 40 0 FFFFFFFF 1

][2 0 0 0 5 Value 0

][3 0 0 0 7 Sig.Err 0

][4 0 0 0 6 Lim H2 0

][5 0 0 0 6 Lim H1 0

][6 0 0 0 6 Lim L1 0

][7 0 0 0 6 Lim L2 0

][8 0 0 0 5 Value 0

][9 0 0 0 8 Printout 0

][10 0 0 0 5 Alarm 0

][11 0 0 0 6 Update 0

][12 0 0 0 8 Disturb. 0

][13 0 0 0 7 Opening 0

][14 0 0 0 7 Closing 0

][15 0 0 0 8 NotClose 0

][16 0 0 0 8 Not Open 0

.....and so on.....

The dump contains all records from 2 to 278. However some of the texts are not longer used and no translation is necessary. The records to be translated are:

record no 2 - 132
 150 - 221
 232 - 233
 238 - 239
 261 - 278

4. Enter the Terminal Window

Edit the file by using a standard text editor for example:

- Enter the **System Configuration Dialog**. Select <**AdvaBuild...**> and thereafter <**Text Editor**>. The file is found in directory /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE.

5. The new dump must be loaded into the target system manually:

LODB SRCE:<file name> for example LODB SRCE:STANDARD

Answer the questions as follows:

- TARGET NOT IN CONFIGURATION MODE. PROCEED? Y
- LOAD MODE: I
- RECEIVER LF = 31
- ENTER RECORD SELECT NODE? N
- LOAD START AT RECORD = 2

6. Take a backup of the RTA Board by selecting <**RTA Board Backup**> in the System Configuration Dialog.

7. Make a Function Restart.

3.2.9.2 System Alarm Texts

An alarm in the System List has the following layout:

* POW SUPP ST	24V I/O supply A faulty	Net 22 Nod 56	96-04-01 14:29:41:425
<i>System Part</i>	<i>System Part Description</i>	<i>Net & Node</i>	<i>Date & Time</i>

Figure 3-7. Alarm layout in the system list

The text row contains:

- System Part text which is a description of in which subsystem the error occurred.
- System Part Description which gives the operator more detail about the alarm.
- Net and node which guides the operator to the correct System Status display.
- Date and time of the alarm.

Translation to a native language must be made by using the database source code facilities for dumping and loading the TEXT concept against the Windows NT environment. The language in the dump depends on the system language in Station Settings. It may be English, German or Swedish as standard. If another language is included in the OS this can of course be used as original language if the system messages have been translated to this language.

A help to translate system messages, the **System List Translator**, is included in the OS.

To use this tool you must first make a logical dump as described below:

1. Enter the **System Configuration Dialog**. Select <**Maintenance...**> and thereafter <**RTA Board Config**>
2. Enter in RTA window
 - EXECFILE INIT:DUSYS

A dump is generated and is stored in directory /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE/. It is called TEXT.LD.

You activate the **System List Translator** as follows:

1. Select **Stations** and open a **Terminal Window**.
2. Enter in the window:
 - SysListTranslator

A list showing all the system messages in the original language is displayed. The list is an extract of the original logical dump. You can not make any selections in this list. It is only used as an overview of all the system messages.

If a translated file exist an overview of these system messages is also displayed.

System List - Overview

message number	parameter	parameters
1220	AF100 STATUS Fatal bus error	Net@1A3@ Nod @2A3@
1221	AF100 STATUS Cable @2I3@ broken	Net @1A3@ Nod @2A3@

Figure 3-8. Extract from the list containing all system messages

A display also shown in which you make the translation, see [Figure 3-9](#).

System List Translator	
<u>F</u> ile	<u>H</u> elp
Message number: 280 PROC I/O ST @3A2@-board error Net@1A3@ Nod@2A3@	Apply Next

Figure 3-9. Dialog box for entering system message text

At start-up the first message (no 280) is displayed. The three text fields to the left contain the original text. In the three empty text fields to the right you enter the text for the native language.

The three text fields display the following text:

- System Part Text
- System Part Description
- Net and Node number.

There may be parameters in the text. The parameter is replaced in runtime by its actual value for example net number. A parameter starts and ends with the sign “@”. The parameters must be written in the same way as in the original version (for example @1A3@).

As the net and node text is the same for all system messages you only have to type it once. It remains in the third text field.

There is a limitation of characters in the text fields. The limit is 12 characters for 8 bit characters for the System Status Text and 28 characters for 8 bit characters for the System Part Description text. Corresponding values for 16 bit characters (for example chinese) are 6 characters for the System Status Text and 14 characters for the System Part Description text.

You can navigate between text fields in a predefined pattern. After entering text in a text field you move the cursor to the next text field by pressing the tab function key on the keyboard.

You must save the entered text for each message. You do this by pressing the **Apply Next** button which also does a step forward in the system message list. that is the next system message is displayed. The text fields to the right are left empty for your input, except the Net and Node text field which you only have to fill in once.

You can also type a message number in the number field and press <CR> to get the text belonging to the message.

The Menu Bar is located in the upper part of the display and consists of pull-down menus for all the features and activities you need to translate the system texts:

File

New	A new empty file is created. The text fields to the right are empty and ready for input.
Open...	A list of all the translated files as well as the original TEXT is displayed. This is only useful if you work with more than one translation. You select which file you want to work with. The text fields to the right contain the translated text. The text fields to the left still contain the text from the original dump. The translated text can be modified.
Save	The translated text is saved and the tool calculates new text pointers. The text pointers are used to position the text correctly in the system list. The text and the pointers are put in an outfile which shall be loaded into the system.
Save As...	Same as Save but you must give a name to the translated file. The maximum length of the name is 8 characters for example FRENCH, DUTCH or SPANISH.
Exit	Terminates the System List Translator. If you have unsaved changes a box is displayed asking you to Save before Exit.

Help

Helps you to use the tool.

The new dump which is created after Save or Save As... is located in the same directory as the original dump that is /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE. The new dump must be loaded into the target system either using a script or manually.

Alternative 1:

Using the script you must use a predefined name on the translated file.

1. Enter the **System Configuration Dialog**. Select **Stations** and open a **Terminal Window**.
2. Enter
 - a. `cd /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE`(You go to a directory)
 - b. `ll` (All files in the directory are listed.)
 - c. `cp XXXX.LD NEW.LD` (This means that you copy the file you have translated and named XXXX to a file called NEW.LD.)

NOTE

You must use capitals for the name and add the suffix .LD.

3. Enter the **System Configuration Dialog**. Select <**Maintenance...**> and thereafter <**RTA Board Config**>
4. Enter in RTA window
 - EXECFILE INIT:LOSYS

Alternative 2:

Making the load manually you do not have to rename your translated file.

1. Enter the **System Configuration Dialog**. Select <**Maintenance...**> and thereafter <**RTA Board Config**>
2. Enter in RTA window
 - LODB SRCE:XXX where XXX is the name of the dump.

NOTE

Max number of characters for the name is 8 and the characters must be capitals.

Answer the questions as follows:

- NOT IN CONFIGURATION MODE PROCEED? Y
- LOAD MODE? I
- RECEIVER LF= 31
- ENTER RECORD SELECT LOAD = N
- LOAD START AT RECORD = 280
- ENTER PROPERTY CHANGE DIALOG? N

3. Take a backup of the RTA Board by selecting <**RTA Board Backup**> in the System Configuration Dialog.
4. Make a Function Restart.

NOTE

The name of the dump in the directory /ABB/Advant/SystemModules/Data/RTA/nodes/RTA/SRCE gets a suffix “.LD”. This suffix shall not be given in **Save As...** or when dumping or loading.

3.2.10 Printer Handling

In an operator station with AdvaCommand version 1.7 for Windows NT, three main types of printouts are used, line printer printouts, postscript printouts and hardcopy printouts. This section describes, for each printout type, which changes that are needed when using a new language. For additional information about printers in an operator station with AdvaCommand version 1.7 for Windows NT, see the *User's Guide AdvaCommand Basic Functions*, section Printer Configuration.

3.2.10.1 Line Printer

Line printer printouts are used for Alarm and Event printouts, Status List, AMPL generated reports and printouts of text files in a standard operator station with AdvaCommand for Windows NT software.

AMPL generated reports uses the printer queues Printer1 and Printer2 and supports characters used in American-English, German and Swedish.

Alarm, Event and Status List printouts uses the printer queues Line and List. But when you use Alarm and Event printout you have to configure the printer to use ISO 8859-1 character set and when you print out from the Status List you should use ASCII.

3.2.10.2 PostScript

Postscript printouts are used by the functions “On-line Builder” and “RTA Board Config.”.

On-line Builder / RTA Board Config

Postscript printouts are handled the same way for both On-line Builder and RTA Board Config. Both functions uses the printer queue OnlineB for its printouts. The printer queue OnlineB is default configured for printouts using a postscript printer. On-line Builder supports the languages American-English, German and Swedish.

On-line Builder and RTA Board Config supports input of text using characters available in American-English, German and Swedish. A consequence of this is that printouts from On-line Builder and RTA Board Config may only include characters from these languages.

3.2.10.3 Hardcopy

No changes are needed for hardcopy printouts. This printout type uses bitmapped screen dumps and is independent of different languages.

3.2.11 RTA- Board Backup

The language dependent dumps on the RTA board are called DATH01.\$LANG where \$LANG is the language variable for example en_US.iso88591 for the English-American version.

The dumps are stored in the directory /ABB/Advant/SystemModules/Data/INIT. The dump contain texts for for example Event and Alarm Handling, Status List and TTD properties.

When introducing a new language you must copy one of the dumps in the directory /ABB/Advant/SystemModules/Data/INIT and name the copy in a predefined way for example sv_SE.iso88591 for the Swedish version

When translating the text for example event text and system messages text you edit the copied file.

3.2.12 Station Backup

To save the additions for localization use **Station Backup**.

Take a backup of the Operator Station Disk Data. Select **Backup of OS Configurable Data**. To add the additions for localization select the files with **Interactive Restore**.

3.3 How to Add and Modify Texts

There is one subsystem defined for customer , named bsCUSTOMER, and one for adaption, named bsADAPTION and one for standard, named bsOS. Each subsystem has a component definition file. These files has the following name and location:

```
/products/data/customer/OS/nls/bsCUSTOMER.def  
/products/data/adaptions/OS/nls/bsADAPTION.def  
/products/data/standard/OS/nls/bsOS.def
```

NOTE

The file bsOS.def is also located under customer and adaption so the system can find modified standard NLS texts under customer and/or adaption

The syntax for the component definition file is:

'subSystem'		
'component'_ 'subSystem'	number	'file name for the message catalog '
'component'_ 'subSystem'	number	'file name for the message catalog '
'component'_ 'subSystem'	number	'file name for the message catalog '

and so on....

Example:

bsCUSTOMER		
processA_bsCUSTOMER	1	processA.cat
processB_bsCUSTOMER	2	processB.cat
specials_bsCUSTOMER	3	Specials.cat

A maximum of 99 (1-99) components may be defined. Our recommendation is to divide the components in such a way that as few message catalogs as possible are used at the same time when a process display is presented. For example divide the components according to the different process sections. For each component a header file is needed that contain mnemonic names for each numerical value. The header file is stored in OS/include/. To create the header file the component file is compiled with the command bsMessageGenerate.

The command syntax for bsMessageGenerate is:

bsMessageGenerate -d file

The syntax for the message files are:

'component'_ 'subsystem'		
'textname'_ 'component'_ 'subsystem'	number	'text'
'textname'_ 'component'_ 'subsystem'	number	'text'
'textname'_ 'component'_ 'subsystem'	number	'text'
and so on....		

Example:

processA_bsCUSTOMER		
StatorCurrent_processA_bsCUSTOMER	1	Stator Current
StatorVoltage_processA_bsCUSTOMER	2	Stator Voltage
StatorCurrent_processB_bsCUSTOMER	101	Stator Current
StatorVoltage_processB_bsCUSTOMER	102	Stator Voltage
SpecialTexts_specials_bsCUSTOMER	201	Special Text
and so on.		

The numbering do not have to be in a sequence but must be in numerical order. It is allowed to group the texts in number groups. In the example above Texts for Process A starts with number 1 and Process B with number 101. Considering future expansions, it is recommended to divide the numbering in message catalogs in groups. A maximum of 999 (1-999) texts may be defined in a message file.

To create the message catalog and the header file that is needed for each message file.

The message file is compiled with the command:

dcNLS file

Message files are normally created for the local language and for en_US.iso88591 (American-English) language. All software uses the text stored in directory en_US.iso88591 if not found elsewhere.

Some examples

- To change a standard NLS text. For example Input Blocked to Inp Bl for an AI: (bsOS is still used because it is a standard NLS text)
 - a. Create a file dcAi.err under adaptions or customer:
dcAi_bsOS
InputBlocked_dcAi_bsOS 3 Inp Bl
 - b. Generate a message catalog and a header file for dcAi.err
dcNLS dcAi.err.
- To create new texts. For example texts to an own display element MOTVAL: (bsADAPTION is used because the text is added under adaptions and it is not a standard NLS text)
 - a. Create a file dcMOTVAL.err under adaptions:
dcMOTVAL_bsADAPTIONS
message1_dcMOTVAL_bsADAPTION 1 Message1
message2_dcMOTVAL_bsADAPTION 2 Message2
 - b. Generate a message catalog and a header file for dcMOTVAL.err
dcNLS dcMOTVAL.err
 - c. Add the message catalog to bsADAPTION.def
bsADAPTION
dcMOTVAL_bsADAPTION 1 dcMOTVAL.cat
 - d. Generate a header file for bsADAPTION.def
bsMessageGenerate -d bsADAPTIONS.def.

3.3.1 Display Builder

The Display Builder is a menu-driven, interactive utility program that you use to plan and design custom graphic displays in the operator station with AdvaCommand for Windows NT software. **This section gives a description of the additions done for the NLS handling in the Display Builder.**

For additional information about the Display Builder, read the *User's Guide AdvaBuild Display Builder*.

3.3.1.1 Display Description

1. Choose **Display Attributes** from the **Tools** menu.

The Display Attributes box is displayed.

Figure 3-10. Display Attributes box

The description text can be defined either as an ordinary string or as an NLS reference. This NLS reference is defined as a symbolic constant in a Text file - dcUser1.err

2. If you choose an ordinary string, enter the description in the Description field, for example:

Overview Feeder.

If you choose an NLS reference, enter the symbolic constant in the Description field like:

OverviewFeeder_dcUser1_bsCUSTOMER

NOTE

The button in front of 'NLSText' must be pressed. Otherwise it will be handled as an ordinary text string.

- 3. If you don't know what NLS references you have, it is possible to search for an NLS reference or create a new. Click on the **Select or Create a NLS-Text** button, after a few seconds a Text Editor Window appears.

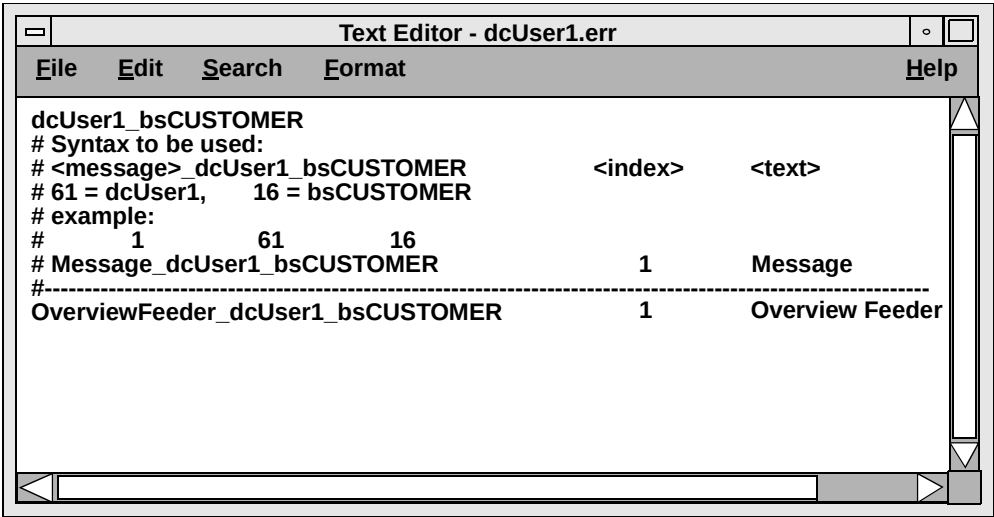


Figure 3-11. Text Editor - dcUser1.err

- 4. Search for an NLS reference. When you find your NLS reference, enter the symbolic constant in the Description field like:

OverviewFeeder_dcUser1_bsCUSTOMER

If you do not find any suitable NLS reference, then you can create a new, for example, an NLS reference for **Pulp Produce**.

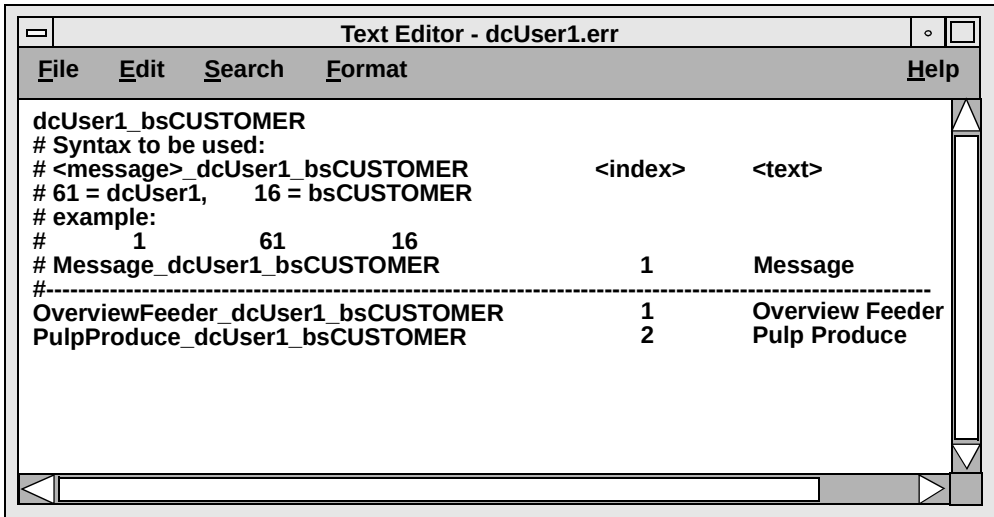


Figure 3-12. Text Editor - dcUser1.err

5. After you add the new NLS reference and save the file dcUser1.err, enter the new symbolic constant in the Description field.

PulpProduce_dcUser1_bsCUSTOMER

6. After editing the new symbolic constant in the NLS-Text field, click **OK** or **Apply**.
When clicking **OK** or **Apply**, a background program - bsMessageGenerate - generates a catalog file and a header file.
7. If a display is installed with an already defined symbol constant, it's very easy to correct a misspelling while working with the Display Builder.
 - a. Click on **Select or Create a NLS-Text** button.
 - b. Correct the misspelling and save the file dcUser1.err.
 - c. Click **OK** or **Apply**.

When you bring up the display again the text should be correct.

3.3.1.2 Dynamic Keys

1. Click on **Define ...** to define dynamic function keys connected to the display.
A new dialog box is displayed.

The screenshot shows a dialog box titled "Dynamic Key Definition". It has several sections: a "Display" field at the top; a list of dynamic keys (D01 through D10); a "Key Label" field; a "NLSText" checkbox with a "Select or Create a NLS-Text" button and a "Help" button; a "Text Font" field; a "Font Color" field with a color swatch; a "Key Type" section with "Base Display" (selected) and "Free" (unchecked) radio buttons; and a "Display Name" field. At the bottom are four buttons: "OK", "Apply", "Cancel", and "Help".

Figure 3-13. Dynamic Key Definition box

2. Click on one of the keys, for example, **D01**, in the list.

The key label text can be defined either as an ordinary string or as an NLS reference. These NLS reference are defined as a symbolic constant in a Text file - dcUser1.err

3. If you choose an ordinary string, enter the key label in the Key Label field, for example,

Report Display

4. If you choose an NLS reference, enter the symbolic constant in the Key Label field like.

ReportDisplay_dcUser1_bsCUSTOMER

NOTE

The button in front of 'NLSText' must be pressed. Otherwise it will be handled as an ordinary text string.

5. If you don't know what NLS references you have, it is possible to search for an NLS reference or create a new. Click on **Select or Create a NLS-Text** button, after a few second a Text Editor Window appears.

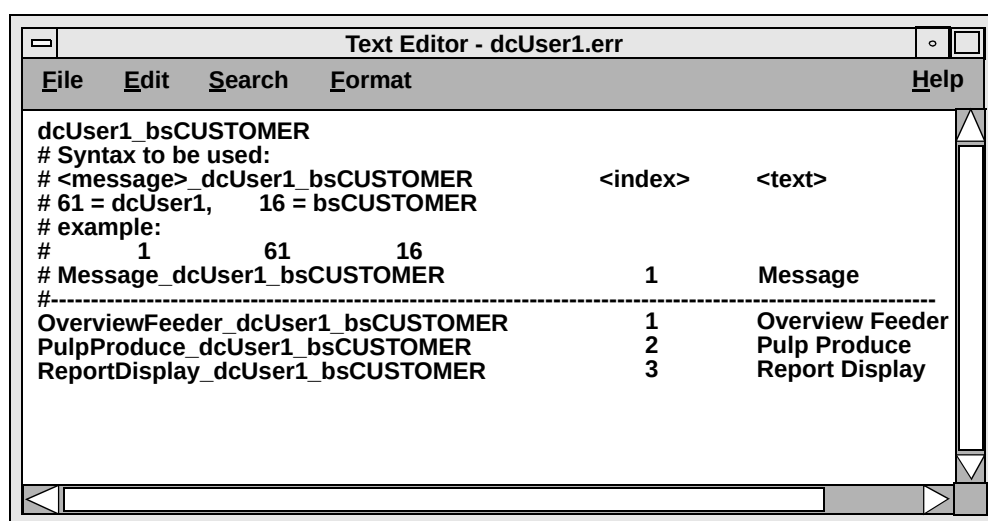


Figure 3-14. Text Editor - dcUser1.err

6. Search for an NLS reference. When you find your NLS reference, enter the symbolic constant in the Description field like.

ReportDisplay_dcUser1_bsCUSTOMER

7. If you do not find any suitable NLS reference, then you can create a new, for example, an NLS reference for **Pump Overview**.

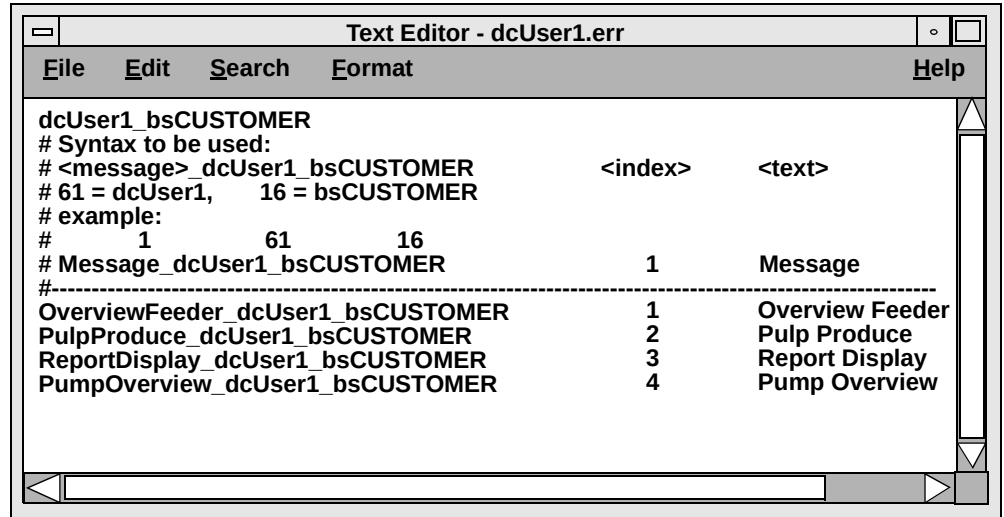


Figure 3-15. Text Editor - dcUser1.err

8. When you have added the new NLS reference and saved the file dcUser1.err, enter the new symbolic constant in the Key Label field.

PumpOverview_dcUser1_bsCUSTOMER

9. After entering the new symbolic constant in the NLS-Text field, click **OK** or **Apply**. When clicking **OK** or **Apply**, a background program - bsMessageGenerate - generate a catalog file and a header file.
10. If a display is installed with an already defined symbol constant, it's very easy to correct a misspelling while working with the Display Builder.
 - a. Click on **Select or Create a NLS-Text** button.
 - b. Correct the misspelling and save the file dcUser1.err.
 - c. Click **OK** or **Apply**.

When you bring up the display again the text should be correct.

3.3.1.3 Display Element

To display the Attributes box for a text display element, do one of the following actions:

1. Double-click on the display element,
2. Select the display element and choose **Display Element Attributes** from the **Tools** menu.

A box is displayed where you can set attributes for the text display element NLSPD01.

The screenshot shows a dialog box titled "NLSPD01ref0". It contains several input fields and buttons. The "Position" section has "X" and "Y" coordinates: X is 53.9312 and Y is 28.8. The "Size" section has "X" and "Y" dimensions: X is 6.09375 and Y is 1.71875. Below these is a "Parameter" section. The "NLS-Text" field is empty. To its left is a checkbox labeled "NLSText" which is checked. To the right of the checkbox is a button labeled "Select or Create a NLS-Text" and a "Help" button. Below the "NLS-Text" field are three more fields: "TextColor" with the value "M16", "TextFont" with the value "FA40", and "Alignment" with the value "1". Below the "Alignment" field is a legend: "1= Left | 2 = Center | 3= Right". At the bottom of the dialog are four buttons: "OK", "Apply", "Cancel", and "Help".

Figure 3-16. Display Element Attributes Box

The NLS-Text can be defined either as an ordinary string or as an NLS reference. This NLS reference is defined as symbolic constant in a Text file - dcUser1.err

3. If you choose an ordinary string, enter the text in the NLS-Text field, for example,

Pump No 1

If you choose an NLS reference, enter the symbolic constant in the NLS-Text field like.

PumpNo1_dcUser1_bsCUSTOMER

NOTE

The button in front of 'NLSText' must be pressed. Otherwise it will be handled as an ordinary text string.

4. If you don't know what NLS references you have, it is possible to search for an NLS reference or create a new. Click on **Select or Create a NLS-Text** button, after a few second a Text Editor Window appears.

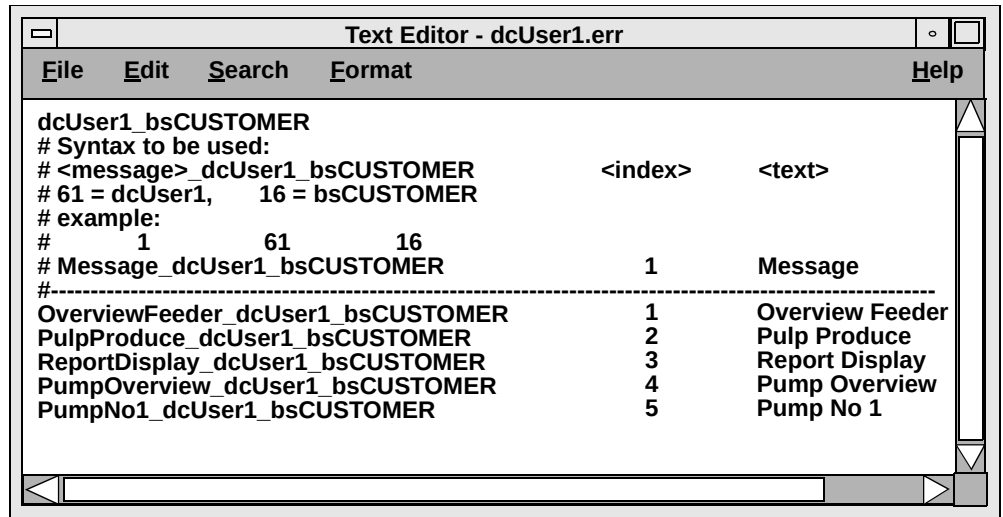


Figure 3-17. Text Editor - dcUser1.err

5. Search for an NLS reference. When you find your NLS reference, enter the symbolic constant in the NLS-Text field like.
PumpNo1_dcUser1_bsCUSTOMER
6. If you do not find any suitable NLS reference, then you can create a new, for example, an NLS reference for **Feeder Speed**.

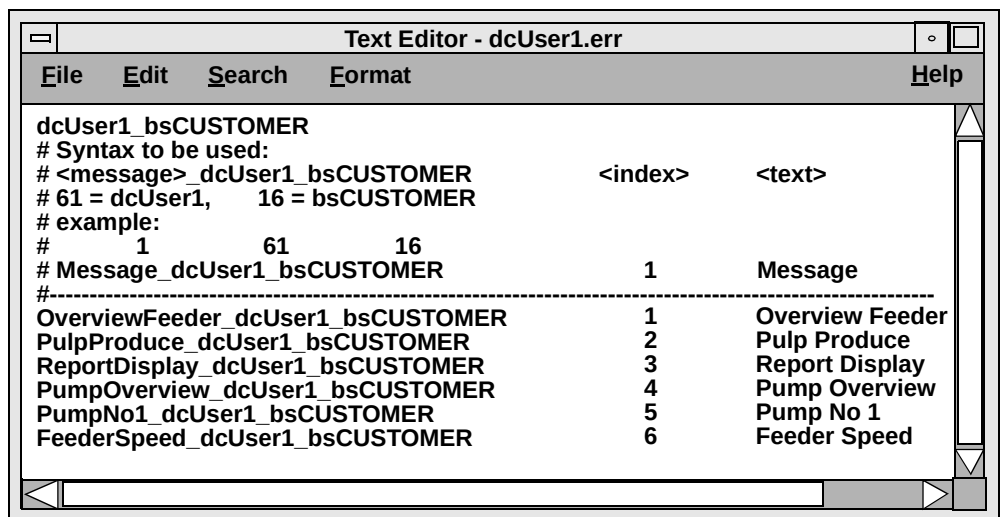


Figure 3-18. Text Editor - dcUser1.err

7. When you have added the new NLS reference and saved the file dcUser1.err, enter the new symbolic constant in the Description field.

FeederSpeed_dcUser1_bsCUSTOMER

8. After editing the new symbolic constant in the NLS-Text field, click **OK** or **Apply**. When clicking **OK** or **Apply**, a background program - bsMessageGenerate - generate a catalog file and a header file.
9. If a display is installed with an already defined symbol constant, it's very easy to correct a misspelling while working with the Display Builder.
 - a. Click on **Select or Create a NLS-Text** button.
 - b. Correct the misspelling and save the file dcUser1.err.
 - c. Click **OK** or **Apply**.

When you bring up the display again the text should be correct.

3.3.2 Process Object Description

Each process object description or other process object text in the database is recognized by its symbolic name, for example **PulpProduce_dcUser1_bsCUSTOMER**, its text "Pulp Produce" or its NLS-index, 16,61,2. Where bsCUSTOMER = 16, dcUser1 = 61 and PulpProduce = 2.

The NLS-index is entered in the process object database and used as a cross-reference link between process object description and the language specific text. The NLS-index is seven (7) characters long and entered into the description field of the database for each process object. The following format is used:

SSCCNNNFree

SS = Subsystem number. Uses 2 characters, legal values are 00-99.

bsADAPTION = 15 and bsCUSTOMER = 16.

CC = Component number. Uses 2 characters, legal values are 00-99.

NNN = Text number. Uses 3 characters, legal values are 000-999.

Free = The remaining text space for Description is 13 characters. These may be used in any way you like, for example English descriptive text.

Example of a process object description using NLS index:

"1661002PulpProduce",

where

16 = *subsystem* bsCUSTOMER

61 = *component* dcUser1

002 = *message* Pulp Produce and

PulpProduce is free English text used in the remaining 13 characters of the process object description.

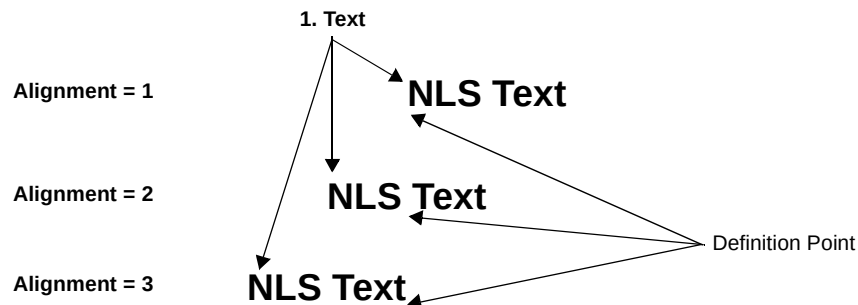
When a message file is created, edited and defined in the component definition file, it must be compiled into a message catalog. As described earlier, the command used is bsMessageInstall.

3.3.3 Display Element and Process Dialogs

A display element for user defined NLS text .

3.3.3.1 Standard Display Element NLSPD01

Presentation:



Behavior:

No	Description	Presentation	Condition	Remarks
1	Text		NLS-Text	The text via ordinary string or NLS reference
		Grey	-	<u>TextColor</u> , <u>TextFont</u>

Configuration:

Parameters	Affects	Options (bold is default)	Remarks
TextColor	1. Text	M16 (Grey), any color	
TextFont	1. Text	FA40 , any font	
Alignment	1. Text	1 = Left, 2 = Center, 3 = Right	

3.3.3.2 Changes in Display Elements

NOTE

This section is only for those who has built or are going to build own designed display elements.

If there is hard coded text in the *.g-file it must be moved to the *.de-file. All hard coded text in the *.de-file must be converted to NLS references with a symbolic constant, ex.

Feeder_dcUser1_bsCUSTOMER

Every NLS reference with a symbolic constant of type Section_dcPid_bsOS or Class_dcAi_bsOS should be converted through a NLSText component, so that the text look-up is made at runtime. Below is shown one old and one new version on a part of a display element.

Old version:

```
var = DynamicVariables
{
    SectionNLSSString = InTerminal
    {
        datatype      = "String10"
        value          = { Section_dcPid_bsOS }
    }
}
```

New version:

```
components = List
{
    TranslatedNLSText = NLSText
    {
        datatype      = "String10"
        in1           = { Section_dcPid_bsOS }
        out1          = "Section"
    }
    var = DynamicVariables
    {
        SectionNLSSString = InTerminal
        {
            datatype    = "String10"
            value        = ::TranslatedNLSText::Section
        }
    }
}
```

By using the method, described in [Section 3.3.2, Process Object Description](#), process object descriptions may be presented on the operator screen in the local language. NLSText is added in the Display Element for translation of the NLS-index, as shown below.

```
TranslatedAttribute = NLSText
{
    datatype      = "String28"
    in1           = ::components::AI::DESCRIPTION
    out1          = "DESCRIPTION"
}
```

Even other text in the object database can be presented in local languages. For example for Motcon the text for the properties PRES_A, PRES_B, IND2_10_PRES_TEXT, IND2_11_PRES_TEXT, IND2_12_PRES_TEXT, IND2_13_PRES_TEXT, IND2_14_PRES_TEXT and ACT_PRES_TEXT.

Below is shown one old and one new version on a part of display element MOTGR01.

Old version:

```
var = DynamicVariables
{
    Description = InTerminal
    {
        datatype = "String28"
        value     = ::components::MOTCON::DESCRIPTION
    }
    ActPresTxtString = InTerminal
    {
        datatype = "String20"
        value     = ::components::MOTCON::ACT_PRES_TEXT
    }
}
```

New version:

```
components = List
{
    TranslatedAttribute = NLSText
    {
        datatype = "String28"
        in1       = ::components::MOTCON::DESCRIPTION
        in2       = ::components::MOTCON::ACT_PRES_TEXT
        out1      = "DESCRIPTION"
        out2      = "ACT_PRES_TEXT"
    }
    var = DynamicVariables
    {
        Description = InTerminal
        {
            datatype = "String28"
            value     = ::TranslatedAttribute::DESCRIPTION
        }
        ActPresTxtString = InTerminal
        {
            datatype = "String20"
            value     = ::TranslatedAttribute::ACT_PRES_TEXT
        }
    }
}
```

Object Display

The datatype for Display Description is changed from String to NativeString, so that the text look-up is made at runtime. Below is shown the old and the new part for Description from the file dcAiObjectDisplay.ds.

Old version:

```
Description  = Property
{
    runtimeAccess = "READ"
    value = { ObjectDisplay_dcMMITEXT_bsOS }
    datatype      = "String"
    factoryAccessible = true
}
```

New version:

```
Description  = Property
{
    runtimeAccess = "READ"
    value = { ObjectDisplay_dcMMITEXT_bsOS }
    datatype      = "NativeString"
    factoryAccessible = true
}
```

3.3.3.3 Changes in Process Dialogs

NOTE

This section is only for those who has built or are going to build own designed process dialogs.

The texts on the dynamic keys are moved from the <file.pcd> to the <file.d>. So we will not have any text dependencies in the <file.pcd>.

Example: In dcAoDialogKeys.pcd all labelstring are removed:

```
labelString = "\ENAcknowledge_dcAo_bsOS";
```

In the <file.d> a new devent called 'dynKeyText' is added. This devent contain all the texts (for example labelString) for the dynamic keys. This devent will be sent when a new Key Group is managed.

Example: When the Key Group with the dynamic keys keyboard is brought up in dcAoDialog.d

```
keyboardKeys does
-- tu_printf( "... keyboardKeys starts \n" );
...
activexWidget.managed := false;
activexWidget := topAo->keybKeys;
this.sendDEvent(dynKeyText,0);
activexWidget.managed := true;
```

```
...
-- tu_printf("--- keyboardKeys ends \n");
end
...
dynKeyText does
-- tu_printf("... dynKeyText starts \n");
i      : integer;
keyName : string;
tmpWidget : widget;
if (activeWidget = keybKeysKeyGroup) and (maxDialog = true)
then
if NLS_keybKeysKeyGroup.count = 0 then

    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(Increase_dcAo_bs0S),1);
    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(MandK_dcAo_bs0S),2);
    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(AutoDK_dcAo_bs0S),3);
    NLS_keybKeysKeyGroup.insert("",4);
    NLS_keybKeysKeyGroup.insert("",5);
    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(Decrease_dcAo_bs0S),6);
    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(Acknowledge_dcAo_bs0S),7);
    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(ObjectDisplay_dcAo_bs0S),8);
    NLS_keybKeysKeyGroup.insert(
        uiGetBsMessage(ObjectTrend_dcAo_bs0S),9);
    NLS_keybKeysKeyGroup.insert("",10);
end if;
i := 1;
while (i < 11) do
    keyName := "db" + (string)i;
    tmpWidget := activeWidget->(keyName);
    tmpWidget.labelString := NLS_keybKeysKeyGroup[i];
    i := i + 1;
end while;
...
-- tu_printf("--- dynKeyText ends \n");
end
```

3.4 Configuration/Application Building Menus

Refer to the manuals for the software. See [Section 1.5, Related Documentation](#).

Chapter 4 Runtime Operation

4.1 Operating Overview

The translation is normally made on the parts that concern the operator. The translated texts will be presented to the operator when running the system. Some texts are not translated and not all the configuration tools are translated. For configuration of the system, normally you need some knowledge in English. Some texts residing in the AC 400 data base are not possible to translate.

4.2 Runtime Operation Menus

The texts in the system are fetched and presented in runtime. It is preferred to present the NLS text at runtime as:

- Texts (for example message catalogs) can be updated and the result is taken into operation without the need to reconvert displays, display elements or process dialogs.
- It is no need to keep object files for several languages in parallel.

Chapter 5 Maintenance

5.1 Backup/Restore Procedures

For backup and restore procedures, refer to the *Release Notes* for more information.

Appendix A Text Files

A.1 dcBasic

The text in the text file dcBasic.err comes mostly from the basic functions in the Advant Operator Station. Here follows a description of the texts from the basic functions:

Table 1-1. Texts from basic functions

No.	Basic Text	Comments
1	Hardcopy: Alternative printer is used for printout.	Alternative printer is used for printout. Hardcopy error = 1
2	Display Name:	Prompt: For display request
3	Config. error: Can not create base display of display type %T.	Base display carrier could not be created
4	System error: Can not create display request state.	Failed to create display request state
5	System error: Can not create overlap display %T for display printout.	Failed to create display state
6	System error: Can not create overlap display %T.	Failed to create overlap display
7	Can not find dcBasicDisplayKeys.	
8	The display %T is not allowed to be presented as an overlapping display.	The display is not allowed to be presented as an overlapping display - isOverlapDisplay = NIL
9	The display %T is not found.	Display factory not found, = nil
10	The overlap display %T is not found.	Display factory not found, = nil
11	There is no help available for this display.	The display doesn't define any help display
12	Hardcopy: Illegal file name.	Illegal file name. Hardcopy error = 3
13	System error: Illegal screen number for display change: %T.	Slave screen number out of range
14	InstalledInOptions	Labelstring on dyn key D7, BASIC DISPLAY
15	The D1 key is not valid.	Dyn keys, the display doesn't define the key
16	The D2 key is not valid.	Dyn keys, the display doesn't define the key
17	The D3 key is not valid.	Dyn keys, the display doesn't define the key
18	The D4 key is not valid.	Dyn keys, the display doesn't define the key
19	The D5 key is not valid.	Dyn keys, the display doesn't define the key
20	The D6 key is not valid.	Dyn keys, the display doesn't define the key
21	The D7 key is not valid.	Dyn keys, the display doesn't define the key
22	The D8 key is not valid.	Dyn keys, the display doesn't define the key
23	The D9 key is not valid.	Dyn keys, the display doesn't define the key
24	The D10 key is not valid.	Dyn keys, the display doesn't define the key
25	Max %T overlapping displays are allowed.	Max no of overlaps already presented

Table 1-1. Texts from basic functions (Continued)

No.	Basic Text	Comments
26	The Menu key is not valid.	The MENU key doesn't exist.
27	The NEXT DIALOG BLOCK key is not valid.	The NEXT DIALOG BLOCK key, the display doesn't define it
28	Config. error: A display link may not have object name set to "NIL".	Display Element with Action = dialog and with object name set to NIL
29	The AUTO key is not valid.	The AUTO key, the dialog doesn't use it
30	The DIALOG ENTRY key is not valid.	The DIALOG ENTRY key doesn't exist
31	The E1 key is not valid.	The E1 key, the dialog doesn't use it
32	The E2 key is not valid.	The E2 key, the dialog doesn't use it
33	The LARGE DECREASE key is not valid.	The LARGE DECREASE key, the dialog doesn't use it
34	The LARGE INCREASE key is not valid.	The LARGE INCREASE key, the dialog doesn't use it
35	The MAN key is not valid.	The MAN key, the dialog doesn't use it
36	The OFF key is not valid.	The OFF key, the dialog doesn't use it
37	The ON key is not valid.	The ON key, the dialog doesn't use it
38	The SMALL DECREASE key is not valid.	The SMALL DECREASE key, the dialog doesn't use it
39	The SMALL INCREASE key is not valid.	The SMALL INCREASE key, the dialog doesn't use it
40	No default dialog found for object type %T.	No dialog set and no default dialog found in Object Table
41	The object type %T has no object display defined.	ObjectDisplay, ObjectCloseup name in Object Table = ""
42	Object display: No object identity is found.	NoOID: OID = nil
43	Object display: Can not find object type %T.	The object type is not found in Object Table
44	The object type %T has no object trend display defined.	The property "Object Trend Display" = "" for the type
45	There is no previous selected object.	PrevObject: previousOID = nil
46	None of the printer queues exist in the system.	None of the printer queues exist in the system.Hardcopy error = 2
47	Object display: Failed to start subscription for object %T.	Object Display Element = nil, no display element to subscribe on
48	Name select is active. Object name is %T.	Another object name is checked if it can be resolved
49	Object name %T is not found.	Object name can not be resolved
50	Object not found.	Cursor analyze with no element or no element.oid found
51	Overlap Display Name:	Prompt: For display request
52	The PREVIOUS DIALOG LEVEL key is not valid.	The PREVIOUS DIALOG LEVEL key, the display doesn't define it
53	PRINT OUT DISPLAY -	Prompt: with factory name and description
54	Hardcopy: Screendump failed.	Screendump failed. Hardcopy error other than 1, 2 and 3
55	Software versions	Labelstring on dyn key D8, BASIC DISPLAY
56	System error: Can not create object closeup request state.	Failed to create objectRequest state
57	System error: Can not create object request state.	Failed to create objectRequest state

Table 1-1. Texts from basic functions (Continued)

No.	Basic Text	Comments
58	The indicated object %T is not found.	Select, oid not found for a selected process object
59	The Trend Closeup key is not valid.	TrendCloseupKey, there is no such key defined.
60	The object %T can not be selected.	Select, process dialog set to NIL in element, i.e no dialog possible
61	Config. error: Can not create overlap display of display type %T.	Failed to create overlap display carrier state
62	<All>	Show all displays string in DisplayMenu.
63	There are no object displays defined for a display link.	Object display requested for a display link
64	Libraries	Title string in DisplayMenu
65	NAME	Title string in DisplayMenu
66	Base	Base xxxxxxxx
67	Overlap	Overlap xxxxxxxx
68	Description	Title string in DisplayMenu
69	<System>	xxxxxxxxxx
70	Previous selected object is the same as the currently selected.	Previous object is the same as currently selected
71	System error: Can not create display %T.	Creation of display state failed
72	%T is not allowed to be presented as base or overlap.	Base and overlap, nothing is allowed
73	Select, Object Name:	Prompt: For object request
74	Closeup, Object Name:	Prompt: For closeup request
75	System error: Can not create display type for display printout.	Failed to create carrier state for display printout
76	BASICDISPLAY	Prompt: BASICDISPLAY, display name
77	Operator Station Version Display	Prompt: BASICDISPLAY, display description
78	Display Menu	Prompt: DisplayMenu, display name
79	Key in Max Dialog position	Key in Max Dialog position
80	Key in Min Dialog position	Key in Min Dialog position
90	Nail this display	
91	N	
92	Ctrl Meta<Key>N	
93	Remove nail	
94	R	
95	Ctrl Meta<Key>R	

A.2 dcDialogMsg

The text in the text file dcDialogMsg.err comes from the dialog for each process object.

Below is a description when the text will appears, some text are self-evident when they appears.

Table 1-2. Description when the texts will appear

No.	Dialog Message Text	Comments
1	%T is selected by another user	Object is already selected by another user
2	%T is not allowed to be selected	Object is not allowed to be selected
3	Invalid select status on %T	Invalid select status on selected Object
4	%T could not be selected	Object could not be selected
5	Selection time-out on %T	Selection timeout on selected Object - Not used yet.
6	Dialog time-out on %T	Selection timeout on Object if the dialog is inactive more than 2 minutes
7	Prevented authority to operate on %T	Prevented authority to operate on selected Object
8	Warning! Outside authorized area to operate on %T	Warning! Outside authorized area to operate on selected Object
9	Entered data is not valid. Must be a number	Entered value in CommandLine is of incorrect format
10	Output or Setpoint is not active	Output or Setpoint is not active for selected Object, GENCON, GENUIS, PIDCON and PIDCONA
11	Output or Ratio is not active	Output or Ratio is not active for selected Object, RATIOSTN
12	There are no more dialog blocks on this level	There are no more dialog blocks on this level - Appears when operator hit the "Next Block" key.
13	This is the top dialog level	This is the top dialog level - Appears when operator hit the "Previous Level" key.
14	Order not allowed in MIN Dialog	Order not allowed in MIN Dialog
15	Entered year value is not valid	Entered year value is not valid for DAT
16	Entered month value is not valid	Entered month value is not valid for DAT
17	Entered day value is not valid	Entered day value is not valid for DAT
18	Entered hour value is not valid	Entered hour value is not valid for DAT
19	Entered minute value is not valid	Entered minute value is not valid for DAT
20	Entered second value is not valid	Entered second value is not valid for DAT
21	%T could not be selected, object not active	Object could not be selected, object not in active group
22	%T could not be selected, object not bound	Object could not be selected, object not bound to server
23	The object could not be selected, the object has no dialog	The object could not be selected, the object has no dialog. It's only a dialog skeleton defined for object type GU2, GU3, GU4, GU5, GU6, MMCX36, MMCX37 and MMCX38.
24	Entered data is not valid. Must be an integer	Entered data is not valid. Must be an integer for object type GU1.
25	Value out of range for INTW. Input ignored.	Entered value greater than 32767 or less than -32768

A.3 dcMMITEXT

The text in the text file dcMMITEXT.err comes mostly from the MasterPiece or the Advant Controller. It is texts from the Object Handler, TTD and System Status. Below is a description when the text will appears, some text are self-evident when they appears.

The text no. 51 -140 and 1051 - 1080 is used by the Object Handler.

The text no. 150 - 174 and 250 - 299 is used by the TTD.

The text no. 300 -310 is used by the System Status.

Here follow a description of the texts from the Object Handler:

Table 1-3. Description from Object Handler

No.	MMI Text	Comments
51	Illegal operation code.	Illegal operation code in order request.
52	Illegal operation property.	Illegal operation property in order request
53	Object not selected (or not implemented).	Object not implemented or not selected
54	Operation is not allowed when process update is not blocked.	Process updating not blocked
55	Limit out of range. Input ignored.	Limits out of range
56	Value out of range. Input ignored.	Value out of range
57	Limit not used. Input ignored.	Requested limit not used, for AI
58	Manual orders blocked. Command ignored.	Manual orders are blocked
59	Blocking of Integration not allowed. Command ignored.	Block of integration only allowed as PC-input for PIDCON
60	Blocking of Derivation not allowed. Command ignored.	Block of derivation only allowed as PC-input for PIDCON
61	Manual mode selection not allowed.	Manual mode not allowed
62	Auto mode selection not allowed.	Auto mode not allowed
63	E1 mode selection not allowed.	E1 mode not allowed
64	E2 mode selection not allowed.	E2 mode not allowed
65	E3 mode selection not allowed.	E3 mode not allowed
66	Setpoint is tracking. Input ignored.	Setpoint is tracking for PIDCONA and PIDCON
67	Output value can only be changed in manual mode.	Outsignal can only be changed in manual mode for PIDCONA, PIDCON and RATIOSTN
68	Output value is connected to external reference.	Outsignal is connected to external reference for PIDCONA, PIDCON and MANSTN
69	Three point control not implemented. DZ not used.	Deadzone is not implemented for PIDCON

Table 1-3. Description from Object Handler (Continued)

No.	MMI Text	Comments
74	Object not selected. Command ignored.	Object not selected
78	Command not allowed in this mode. Command ignored.	Wrong mode for this operation to be performed
79	Object already blocked.	Object is already blocked
80	Object not blocked.	Object is already deblocked
81	Object already in Manual mode.	Object in manual mode
82	Object already in Auto mode.	Object in auto mode
83	Board already in service.	Already in service
84	Board already out of service.	Already out of service
85	No instances connected to board.	No objects connected to interface board for AO and DO-interface
86	Ratio connected to external reference.	Ratio is connected to external reference for RATIOSTN
97	Object already in hold mode.	Already in hold mode for SEQUENCE
98	Object not implemented.	Object is not implemented
99	Step conditions not fulfilled.	Conditions for stepping are not fulfilled
101	Instance does not exist or board not implemented.	Record is not found OR interface is not implemented or not found
102	Board out of service.	Interface is out of service for AI, AO, DI and DO
103	Hardware error reported.	Hardware error for DI
104	Output temporary locked. Try again.	The record is locked while physical output is done for AO and DO
105	Output failed.	Output failed for AO and DO
106	Attempt to read input failed.	Converter type is out of range for AI
107	Illegal data type.	Illegal datatype in format, i.e. external reference, tag or format reference for DAT
108	Concept/property not found in logical file 1.	Concept/Property name not found in LF 1 for DAT
109	The sequence is not running. Command ignored.	Sequence is not running for SEQUENCE
110	Event print already blocked.	Event print already blocked for SEQUENCE
111	Event print already deblocked.	Event print not blocked for SEQUENCE
112	Event proc. already blocked.	Event processing is already blocked for SEQUENCE

Table 1-3. Description from Object Handler (Continued)

No.	MMI Text	Comments
113	Event proc. already deblocked.	Event processing not blocked for SEQUENCE
114	Limit is not active for this object.	MAX_LIM or MIN_LIM is not used for AO
115	The input is not calculated. Command ignored.	Attempt to change calc_value for a non-calc. input for AI or DI
116	Attempt to change Max limit to be less than Min limit.	New MAX_LIM value is less than actual MIN_LIM value, order is not performed
117	Attempt to change Min limit to be greater than Max limit.	New MIN_LIM value is greater than actual MAX_LIM value, order is not performed
118	Flag in MORD_ALLOWED is not true. Command is not performed.	Order is not performed as corresponding flag in MORD_ALLOWED is not set for GENOBJ
119	Flag in VALUE_ALLOWED is not true. Command is not performed.	Value change order is not performed as corresponding flag in VALUE_ALLOWED is not set for GENOBJ
120	Object is selected by another operator. Command is not performed.	Order is not performed as actual object is selected by another operator (network,node,mmi) for DAT
121	Value out of range.	Value out of range
122	Attempt to change OUT max to be < OUT min.	New OP_MAX value is less than actual OP_MIN value or out of range of (0,100), order is not performed for GERNOBJ
123	Attempt to change OUT min to be > OUT max.	New OP_MIN value is greater than actual OP_MAX value or out of range of (0,100), order is not performed for GERNOBJ
124	Attempt to change SP max < SP min or out of range of MV limits.	New SP_MAX value is less than actual SP_MIN value or out of range of (MV_MIN,MV_MAX), order is not performed, GERNOBJ
125	Attempt to change SP min > SP max or out of range of MV limits.	New SP_MIN value is greater than actual SP_MAX value or out of range of (MV_MIN,MV_MAX), order is not performed
126	Command is not performed. Command blocked by ORD_BLK in the database.	Value change order is not performed as corresponding flag in ORDER_BLOCK is not set for GROUP_ALARM
127	Change of Group Type is not allowed when members are connected to the group.	Value change order is not performed as system not is in configuration mode or memb_count isn't 0 for GROUP_ALARM
128	Flag in ORD_BLK is true. Command not performed.	Value change order is not performed as corresponding flag in ORD_BLK is set for TEXT_DATA
129	Attempt to change dispmax > dispmin. Command is not performed.	Attempt to change DISPMAX > DISPMIN, order is not performed for TEXT_DATA
130	Max no. of connections exceeded. Command is not performed.	The groupmember is already connected to seven groups, order is not performed for GROUP_ALARM
131	Object not in redundancy use or not in service. Command is not performed.	Object not in redundancy use
132	Interface has fatal error status. Command is not performed.	OBJ_INTERFACE has fatal error
133	Board 2 has error status. Command is not performed.	OBJ_INTERFACE: Board 2 has error status
134	Board 1 has error status. Command is not performed.	OBJ_INTERFACE: Board 1 has error status

Table 1-3. Description from Object Handler (Continued)

No.	MMI Text	Comments
135	Member already connected to this group. Command is not performed.	Member already connected to this group, order is not performed for GROUP_ALARM
136	Property already set. Command is not performed.	Value already set for AI
137	Property already reset. Command is not performed.	Value already reset for AI
138	Not supported function for this object type.	Not supported function for this object type
1051	Frequency limit must be greater than zero.	Frequency limit must be greater then zero
1052	The level must be greater than zero and less than or equal to 50%.	The level must be greater then zero and less than or equal to 50%
1053	Mode BAL prevents start of autotuning.	Mode BAL prevents the start of autotuning
1054	Mode MAN CLAMPED prevents start of autotuning.	Mode MAN CLAMPED start of prevents autotuning
1055	Mode MAN FORCED prevents start of autotuning.	Mode MAN FORCED start of prevents autotuning
1056	AI ERROR prevents start of autotuning.	AI ERROR start of prevents autotuning
1057	Autotuning is not in verify phase, prevents CONTINUE TUNE	Autotuning is not in VERIFY phase, prevents CONTINUE TUNE
1058	RESTORE before you CONTINUE TUNE, or use START TUNE.	RESTORE before you CONTINUE TUNE, or use START TUNE
1059	Issue START TUNE, there is nothing to CONTINUE TUNE from.	Issue START TUNE, there is noting to CONTUNE TUNE from
1060	Time-out counter has been restarted, due to CONTINUE TUNE.	Time-out counter has been restarted, due to CONTINUE TUNE
1061	You cannot do ACCEPT & SAVE until the EXCITE or VERIFY phase.	You cannot do ACCEPT & SAVE until the EXCITE or VERFY phase
1062	Since not autotuning ACCEPT & SAVE gave only SAVE.	Since not autotuning ACCEPT & SAVE gave only SAVE.
1063	Autotuning stopped. Present setting unchanged.	Autotuning stopped. Present setting unchanged. Noting saved.
1064	Since not autotuning STOP & RESTORE gave only RESTORE.	Since not autotuning STOP & RESTORE gave only RESTORE
1065	Wait until autotuning is finished before ADAPTATION ENABLE.	Wait until autotuning is finished before ADAPTATION ENABLE
1066	Issue START TUNE before ADAPTATION ENABLE.	Issue START TUNE before ADAPTATION ENABLE
1067	Autotuning prevents SAVE, use ACCEPT & SAVE instead.	Autotuning prevents SAVE, use ACCEPT & SAVE instead
1068	Autotuning prevents RESTORE, use STOP & RESTORE if desired.	Autotuning prevents RESTORE, use STOP & RESTORE if desired
1069	Present setting unchanged. There was no setting to RESTORE.	Present setting unchanged. There is no setting to RESTORE

Table 1-3. Description from Object Handler (Continued)

No.	MMI Text	Comments
1070	Value can only be changed when adaptation and autotuning are off	Value can only be changed when adaptation and autotuning is off
1071	STOP & RESTORE active autotuning, before you START TUNE.	STOP & RESTORE active autotuning, before you START TUNE
1072	Warning: There are parameter sets with invalid sampling rate (TS).	Warning: there are setting with invalid sampling rate.
1073	Autotuning is disabled at PC element.	Autotuning is disabled at PC element.
1074	Value can only be greater than or equal to zero.	Value can only be greater than or equal to zero.
1075	Change of response type is not allowed during verify phase.	Change of response type is not allowed during verify phase.
1076	VERIFY is only valid during the excitation phase of autotuning.	VERIFY is only valid during the excitation phase of autotuning.
1077	There is no aborted setting to retrieve.	There is no aborted setting to retrieve.
1078	You cannot SAVE a setting while waiting for the lost one to be retrieved.	You cannot SAVE a setting while waiting for the lost one to be retrieved.
1079	While retrieving the lost setting RESTORE commands are ignored.	While retrieving the lost setting RESTORE commands are ignored.
1080	Wait until aborted setting has been retrieved.	Wait until aborted setting has been retrieved.

Here follows a description of the texts from the TTD Handler:

Table 1-4. Description from TDD Handler

No.	MMI Text	Reason
250	TTD: The request is correctly performed.	TTD request ok
251	TTD: Function not implemented.	Illegal operation code in ttd request
252	TTD: Log not found.	Log not found
253	TTD: Log not implemented.	Log not implemented
254	TTD: Variable not found.	Variable not found
255	TTD: Variable not implemented.	Variable not implemented
256	TTD: Number of requested variables out of range.	Number of requested variables out of range
257	TTD: Number of requested values out of range.	Number of requested values out of range
258	TTD: Time completely out of TTD's scope	Time completely out of TTD's scope
268	TTD: Illegal logical file number	Illegal reference of logical file number

Table 1-4. Description from TDD Handler (Continued)

No.	MMI Text	Reason
269	TTD: Illegal subscription code (sub type)	Illegal subtype request
270	TTD: Illegal data type (ttd type)	Illegal value type
271	TTD: Illegal value in time specification	Illegal value of time
272	TTD: the log is no user data log	Log not user type
273	TTD: Illegal reference type (ref type)	Illegal reference type
274	TTD: Illegal property number	Illegal property number
275	TTD: Additional information (add inf) missing	No additional info defined
277	TTD: referenced concept not defined in the node	Referenced concept not defined in system
278	TTD: referenced instance not found in the node	Referenced file not found
279	TTD: referenced object or log is not implemented	Referenced file not implemented
280	TTD: referenced variable is not found in the log	Referenced variable not found
281	TTD: referenced variable is not implemented	Referenced variable not implemented
282	TTD: referenced variable is not initiated	Referenced variable not initiated
283	TTD: access to primary logged object failed	Access to primary variable failed
285	TTD: access to primary logged object failed	Failed to get physical address
286	TTD: the log is a backup log	Log is a backup log
287	TTD: program error	Program error in TTD
288	TTD: illegal log reference	Illegal log number
289	TTD: the sample intervals do not match	Illegal sample interval
290	TTD: referenced log is a user data log	Referenced log is user log
291	TTD: referenced log is a backup log	Referenced log is backup log
292	TTD: illegal code for requesting information (geti mode)	Illegal geti_mode
293	TTD: log or log interval must be specified	Neither log_ref nor log_interval was given

INDEX

Symbols

\$LANG 3-7

B

bsADAPTION.def 3-20
bsCUSTOMER.def 3-20
bsGenerate 3-21
bsOS.def 3-20
bsSubSystem.def 3-2

C

C_LANGUAGE 3-6 to 3-7
Component number 3-2

D

Dbuild 3-11
dcCLanguage 3-7
dcSysLanguage 3-7
DisplayTool 3-11

E

Event text 3-11
EVENT_TREAT 3-11 to 3-12

H

Hardcopy 3-19
Header file 3-5

I

Internationalizing 1-5
internationalizing 3-1

K

Keyboard mapping 3-7
KEYMAP 3-6

L

LANG 3-5 to 3-6
Localization 1-5, 3-1

M

Message catalog 3-5
Message files 3-4 to 3-5, 3-21
Message number 3-2

N

NLS 1-5
NLS reference 3-23, 3-25 to 3-29, 3-31
NLS references 3-24
NLSEditor 3-11
NLS-index 3-30

O

OS 3-11

P

Postscript 3-19
Printouts 3-19
Property text 3-11

S

Subsystem number 3-2
System List Translator 3-15
System variables 3-6

T

TEXT 3-11 to 3-12

X

X font 3-10
X-resources 3-11
X-server 3-9

